COMMON DATA FORMAT
TELEMETRY DATA DISTRIBUTION OBJECTS

OVERVIEW

Telemetry data retrieved from flat files or the data warehouse by both users and applications will be delivered via telemetry packet objects. The format for these objects and the data they contain is referred to as the Common Data Format. Data produced by the Front End Processor will have a similar format, to be referred to as the FEP Output Format, with the only difference being the presence of two additional time fields in the Header Object. These two additional time fields will be available to recipients of the real-time FEP output data and by querying the intermediate storage files. They are used for the merge process, but are not archived with the telemetry. An Application Program Interface (API) will be provided to isolate the end user from format concerns. The API will contain all the necessary methods for extracting data and presenting it to the user. This document describes both the Common Data Format and the FEP Output Format with the two items pertaining only to the latter clearly identified.

DETAILS

Each telemetry object consists of a single telemetry data packet. Each packet will consist of a header object containing meta-data about the packet followed by telemetry element objects. Each telemetry element object contains one spacecraft telemetry or derived parameter in both raw and Engineering Unit formats, along with its numeric identifier and associated flags. Descriptions of the Telemetry Data Objects are provided below:

Packet Header Object

Item Name	Bit Size	Type	Item Description
Data Source	8	Short Int	Identifies various characteristics of the data source as shown in Table 1.
Telemetry Format	8	Short Int	Identifies telemetry format. Consecutive integer codes will be assigned to all existing formats as shown in Table 2. Additional codes will assigned for new formats as necessary.
Spacecraft Time	64	FP	Time defined by the vehicle clock count converted to UTC and expressed as modified Astronomical Julian Day. For test data, this field will contain CCS Time when presented to the user if the data capture process was configured to store the data by CCS time during the test.

Item Name	Bit Size	Type	Item Description
****The following two fields occur only in the FEP Output data**			
******NGT Receipt Time	64	FP	UTC of time NGT transmitted the data to CCS expressed as modified Astronomical Julian Day
******CCS Receipt Time	64	FP	UTC of time data was received by CCS expressed as modified Astronomical Julian Day
Number of Elements	16	Int	Number of Telemetry Element Objects in the packet
Element Object			
Numeric ID	16	Int	Unique identifier for each parameter
Raw Value	32	Int	Parameter raw value as stored by the spacecraft
EU Value	32	FP	Parameter value converted to engineering units
Flags	16	Int	Associated flags (see Table 3)

Notes

1. Numeric ID: This is a unique integer assigned to each telemetry point for the life of the mission even if the point is eliminated. The method for mapping the Numeric ID to the 8-character mnemonic name, e.g. CBAT1V, is **TBD**.

2. EU Value: This has been chosen as single precision rather than double precision to save space. While certain applications need double precision values of some telemetry points, particularly for attitude related processing, these applications currently use the raw values. Since there are few of these, doubling the space to accommodate them would not be cost-effective.

3. Data gaps will be indicated by a Boolean pseudo-parameter that indicates data presence. The parameter will change value at the beginning and end of each contiguous data span. The precise definition of a data gap is **TBD**.

Table 1. Definitions of Flags in the CDF Header Data Source Field

BIT	FLAG NAME	DESCRIPTION
0 (lsb)	Spacecraft Data Mode	0 indicates recorded data (ETR/SSR) 1 indicates Real-Time
1	Ground Station Mode	0 indicates direct feed through ground station 1 indicates replay of ground station recorded data
2	CCS Mode	0 indicates operational data 1 indicates test data
3	FEP Mode	0 indicates external data source 1 indicates FEP is replaying pre-recorded data for testing
4	FEP Replay	Set to 1 if data is being supplied in response to an FEP replay request
5	Era	0 if data was captured by CCS 1 if data was converted AEDP/ESS data
6-7	Spare	

Table 2. Telemetry Format Identifier Code Definitions

Code	Format	Code	Format	Code	Format	Code	Format
0	off	37	HF	80	PN	163	U
5	XN	40	FN	81	PF	186	ZN
6	XF	41	FF	82	NSSC-1 Dmp	187	ZF
24	TN	48	C	138	M	192	AN
25	TF	64	YN	145	S	193	AF
36	HN	65	YF	146	D/E	others	spare

Table 3. Definitions of Flag Bits in the CDF Element Flags

BIT	FLAG NAME	DESCRIPTION
0 (lsb)	Corrected Spacecraft Time	Set to 1 if VCC was bad (FEP) and S/C Time has been corrected (archive); also used for D/E format data
1	Quality	Set to 1 if data quality is questionable
2	Limit Low	Set to 1 if out of limits low
3	Limit High	Set to 1 if out of limits high
4	Limit Level	Set to 1 if beyond severe limit (red)
5	Delta Error	Set to 1 if delta limit exceeded
6	EU Conversion Error	Set to 1 for conversion error
7	Reconstructed Point	Set to 1 if this point was added to "changes only" data to construct "all points" data.
8-15	Spare	

API FOR END-USERS

This section illustrates the kinds of methods that will be provided. At the present time, this list is neither complete nor accurate. As the design proceeds, a complete specification of all the methods to be provided will be included here.

Retrieve Element Info:	getID()	returns Numeric ID of the element
	getRaw()	returns raw value of the element
	getEU()	returns EU converted value or raw if there is no EU
	getFlags()	returns all flags related to the element
	dump(...)	returns a preformatted dump of data (overloaded)
Retrieve Packet Info:	getSource()	returns the data source of the packet
	getFormat()	returns the data format of the packet
	getTime()	returns the time tag of the packet
	getNRT()	returns the NGT Receipt time **(FEP Output only)**
	getCCST()	returns the CCS Receipt time **(FEP Output only)**
	getNum()	returns the number of elements in the packet

getElements()	returns the start location of the element
dump(...)	returns a preformatted dump of packet (overloaded)